

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-223096

(43)公開日 平成6年(1994)8月12日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/31		M 7343-5L		
11/10	3 3 0 Q	7313-5B		
// H 0 3 M 13/00		8730-5 J		

審査請求 未請求 請求項の数 7 O L (全 12 頁)

(21)出願番号 特願平5-9334

(22)出願日 平成5年(1993)1月22日

(71)出願人 000001007

キャノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 岩村 恵市

東京都大田区下丸子3丁目30番2号キャノン株式会社内

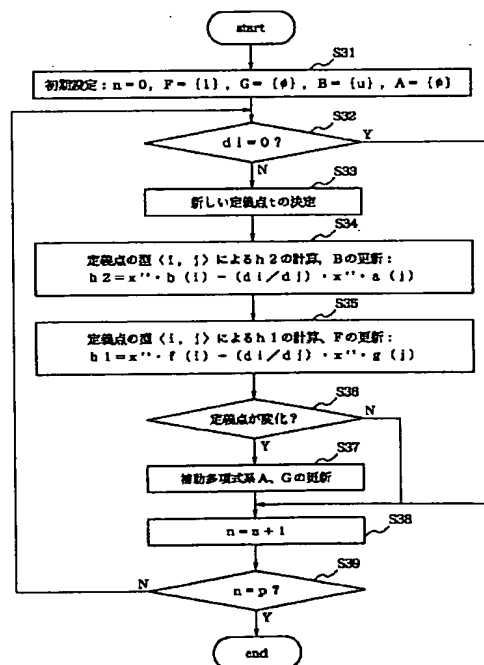
(74)代理人 弁理士 丸島 儀一

(54)【発明の名称】 多項式系導出装置及びその方法

(57)【要約】

【目的】 代数幾何符号の復号のための多次元配列を生成する最小多項式の導出を高速に実行する。

【構成】 与えられた多次元配列を生成する最小多項式系Fを求めるために、多項式系Fを更新する際に、 $df_n^{(1)}$ を直接計算せず、新たに導入した多項式系Bに属する多項式の最高次係数 d_1 を用いて、ステップS34において多項式系Bを更新し、ステップS35において多項式系Fを更新する。



【特許請求の範囲】

【請求項1】 与えられた多次元配列を生成する最小多項式系を求める多項式系導出装置において、
求める第1の多項式系を記憶するための第1多項式記憶手段と、

当該第1の多項式系と異なる第2の多項式系を記憶するための第2多項式記憶手段と、

前記第1及び第2の多項式記憶手段に初期値を設定する設定手段と、

前記第2多項式記憶手段に記憶された第2の多項式系の所定の次数の係数から得られる値を用いて、前記第1多項式記憶手段に記憶された第1の多項式系を更新する第1更新手段と、

前記第2多項式記憶手段に記憶された第2の多項式系を更新する第2更新手段とを有することを特徴とする多項式系導出装置。

【請求項2】 前記第1更新手段による更新動作と前記第2更新手段による更新動作とを並列に実行することを特徴とする請求項1に記載の多項式系導出装置。

【請求項3】 前記第1更新手段と前記第2更新手段とを共通の回路手段によって実現すること特徴とする請求項1に記載の多項式系導出装置。

【請求項4】 前記第2更新手段が、 $b_n^{(i)}$ 、 $b_m^{(j)}$ を前記第2多項式系に属する多項式とし、 d_i 、 d_j を、それぞれ当該多項式 $b_n^{(i)}$ 、 $b_m^{(j)}$ の最高または最低次の係数として、演算

$b_a^{(k)} = z^{r_a} \cdot b_n^{(i)} - (d_i / d_j) \cdot z^{r_t} \cdot b_m^{(j)}$
を行うことを特徴とする請求項1に記載の多項式系導出装置。

【請求項5】 前記第2更新手段が、 $b_n^{(i)}$ 、 $b_m^{(j)}$ を前記第2多項式系に属する多項式とし、 d_i 、 d_j を、それぞれ当該多項式 $b_n^{(i)}$ 、 $b_m^{(j)}$ の最高または最低次の係数として、演算

$b_a^{(k)} = d_j \cdot z^{r_a} \cdot b_n^{(i)} - d_i \cdot z^{r_t} \cdot b_m^{(j)}$
を行うことを特徴とする請求項1に記載の多項式系導出装置。

【請求項6】 与えられた多次元配列を生成する最小多項式系を求める多項式系導出方法において、

求める第1の多項式系を記憶するための第1メモリと、
当該第1の多項式系と異なる第2の多項式系を記憶するための第2メモリとに初期値を設定し、

前記第2メモリに記憶された第2の多項式系の所定の次数の係数から得られる値を用いて、前記第1メモリに記憶された第1の多項式系を更新し、

前記第2メモリに記憶された第2の多項式系を更新することを特徴とする多項式系導出方法。

【請求項7】 前記第1の多項式系の更新動作と前記第2の多項式系の更新動作とを並列に実行することを特徴とする請求項6に記載の多項式系導出方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、ディジタル通信系及びディジタル記憶系において通信路または記憶媒体で受けた誤りを、誤り訂正符号を用いて受信側で自動的に訂正する誤り訂正に関し、特に誤り訂正符号として代数幾何符号を用いる場合の復号において、与えられた多次元配列を生成する最小多項式系を求める多項式系導出装置及びその方法に関する。

【0002】

【従来の技術】 従来、ディジタル通信系及びディジタル記憶系において通信路または記憶媒体で受けた誤りを、受信側で自動的に訂正する誤り訂正符号として、リード・ソロモン(RS)符号やBCH符号がよく知られ、コンパクト・ディスクや衛星通信等において実際に装置化され使用されている(参考文献[1]参照)。

【0003】 しかし、最近では、代数曲線論を駆使した代数幾何符号(参考文献[1]-[4]参照)と呼ばれる符号の研究が盛んに行われている。代数幾何符号は、前述のRS符号やBCH符号をその1クラスとして含む非常に適用範囲の広い符号系であり、従来の符号よりもよい性質を持つ新符号が存在することが徐々に明らかになってきている(参考文献[5][6]参照)。

【0004】 このように優れた符号が発見される中で、恒に問題となっていたのは復号法の問題であり、その効率的な復号アルゴリズムの開発が急がれたが、1989年にJustesenらのグループが一般化Peterson algorithm(参考文献[7]参照)を提案するまで、長い間有効な復号アルゴリズムは全く得られなかった。また、Justesenらは、与えられた有限の大きさを持つ2次元配列を生成する、記憶素子数最小の2次元線形掃選シフトレジスタを求める効率的なアルゴリズムとして、阪田によって提案された阪田アルゴリズム(参考文献[8]参照)を、彼らの一般化Peterson algorithmに応用することで、少ない計算量で高速に誤り位置関数を導出できることを示した。

【0005】 しかし、一般化Peterson algorithmは、その訂正能力に制約があったために、Skorobogatovらはより高い訂正能力を保証するModified decoding algorithm(参考文献[9]参照)を提案した。また、阪田アルゴリズムにはModified decoding algorithmに対しては直接関係のない無駄な演算が多く存在したために、神谷らは、阪田アルゴリズムをModified decoding algorithmに沿った形(参考文献[11][12]参照)に修正したアルゴリズムを1992年に提案した。

【0006】 一方、従来用いられているRS符号やBCH符号の復号法として、Peterson法と、バーレカンブ・マッセイ(BM)法及びユークリッド(Eu)法がよく知られている。前述の一般化Peterson algorithmはRS符号の復号に用いられるPeterson法の拡張になっている。また阪田アルゴリズムは2次元BM法とも呼ばれ、

RS符号の復号に用いられるBM法（以後、1次元BM法と呼ぶ）の拡張になっていることが知られている。

【0007】

【発明が解決しようとしている課題】しかしながら、RS符号の復号に用いられるEu法（以後、1次元Eu法と呼ぶ）の拡張に相当するアルゴリズムは、これまでに考えられていなかった。

【0008】また、阪田は、2次元BM法の拡張として多次元BM法（参考文献[13]参照）も提案したが、それに対応する多次元Eu法もまた、考案されていなかった。

【0009】そこで、本発明では、1次元Eu法の拡張に相当する2次元Eu法を提案し、更に、この2次元Eu法を拡張して、多次元BM法に対応する多次元Eu法を提案することを目的とする。

【0010】

【課題を解決するための手段】上記課題を解決するために、本発明では、与えられた多次元配列を生成する最小多項式系を求める多項式系導出装置に、求める第1の多項式系を記憶するための第1多項式記憶手段と、当該第1の多項式系と異なる第2の多項式系を記憶するための第2多項式記憶手段と、前記第1及び第2の多項式記憶手段に初期値を設定する設定手段と、前記第2多項式記憶手段に記憶された第2の多項式系の所定の次数の係数から得られる値を用いて、前記第1多項式記憶手段に記憶された第1の多項式系を更新する第1更新手段と、前記第2多項式記憶手段に記憶された第2の多項式系を更新する第2更新手段とを具える。

【0011】

【作用】設定手段により、求める第1の多項式系を記憶するための第1多項式記憶手段及び当該第1の多項式系と異なる第2の多項式系を記憶するための第2多項式記憶手段に初期値を設定し、前記第2多項式記憶手段に記憶された第2の多項式系の所定の次数の係数から得られる値を用いて、第1更新手段によって前記第1多項式記憶手段に記憶された第1の多項式系を更新し、第2更新手段によって前記第2多項式記憶手段に記憶された第2の多項式系を更新する。

【0012】【参考文献】

[1] 今井：“符号理論”，電子情報通信学会

[2] V. D. Goppa: “Codes on algebraic curves”, Soviet Math. Dokl., 24, pp. 170-172, 1981.

[3] V. D. Goppa: “Algebraico-geometric codes”, Math. U. S. S. R. Izvestiya, vol. 21, no. 1, pp. 75-91, 1983.

[4] V. D. Goppa: “Geometry and Codes”, Kluwer Academic Publishers, 1988.

[5] M. A. Tsfasman and S. G. Vladut: “Algebraic-geometric codes”, Kluwer Academic Publishers, 1991.

[6] 三浦：“ある平面曲線上の代数曲線符号”，Prepri

nt.

[7] J. Justesen, K. J. Larsen, E. Jensen, A. Havemose and T. Hoholdt: “Construction and decoding of a class of algebraic geometry codes”, IEEE Trans. Inform. Theory, vol. 35, no. 4, pp. 811-821, July 1989.

[8] 阪田：“与えられた2次元配列を生成する2次元線形掃選シフトレジスタの合成”，信学論(A)，vol. J-70A, pp. 903-910, 1987.

[9] A. N. Skorobogatov and S. G. Vladut: “On the decoding of algebraic geometric codes”, IEEE Trans. Inform. Theory, vol. 36, no. 5, pp. 1051-1060, Sep. 1990.

[10] 三浦：“代数曲線符号の一般的な復号法(1), (2)”，信学技法，vol. IT89, no. 452, IT89-91, 92, pp. 61-70, pp. 71-80, 1990.

[11] 神谷，三浦：“ある種の代数曲線符号に関する Modified decoding algorithmへの Sakata algorithm の応用について”，信学技法，vol. IT91, no. 435, IT91-96, pp. 47-54, 1992.

[12] 神谷，三浦：“ある種の代数曲線符号に関する再帰的な復号アルゴリズム”，信学技法，vol. IT91, no. 505, IT91-116, pp. 89-96, 1992.

[13] S. Sakata: “Extension of the Berlekamp-Massey algorithm to N dimensions”, Information and Computation, vol. 84, pp. 207-239, 1990

【0013】

【実施例】

【説明の概要】以下では、まず1次元Eu法の拡張に相当する2次元Eu法を提案する。

【0014】そして、この2次元Eu法が、2次元BM法である阪田アルゴリズムやその修正アルゴリズムである神谷アルゴリズムと等価な結果を出力することを示す。更に、2次元Eu法を拡張し、多次元BM法に対応する多次元Eu法を提案する。そして、本発明による多次元Eu法が、装置化や高速化において、従来の多次元BM法と異なる効果をもつアルゴリズムであることも示し、その違いを明らかにする。

【0015】従来知られた多次元BM法のアルゴリズムは、図2に示す構造を持つ。一方、本発明による多次元Eu法のアルゴリズムは、図3に示す構造を持つ。多次元BM法と多次元Eu法の違いは、前者がステップS22において $df_n^{(1)}$ を直接計算し、ステップS25においてその値を用いて多項式系Fを更新するのに対し、後者では、 $df_n^{(1)}$ を直接計算せず、新たに導入した多項式系Bに属する多項式の最高次係数 d_i を用いて、ステップS34において多項式系Bを更新し、ステップS35において多項式系Fを更新する点である。

【0016】上記従来の多次元BM法が、与えられた多次元配列uを生成する最小多項式系Fを導くことは、前記文献[8]，[13]に証明されている。よって、ここでは本発明による多次元Eu法が、多次元BM法と同じ多項

式系Fを導くことを以下に示す定理1～定理3により証明することによって、図3の多次元Eu法から得られる多項式系Fも、与えられた多次元配列uを生成する最小多項式系であることを示す。以下、従来知られた2次元BM法及び多次元BM法を例にとり、本発明による多次元Eu法を具体的に説明する。

【0017】〔実施例1〕まず、以下の準備（用語の説明）の後に、阪田アルゴリズムを示す。このアルゴリズムが、与えられた有限2次元配列uを生成する、記憶素子数最小の2次元線形帰還シフトレジスタを合成する多項式系Fを導出することの証明は、前記文献[8]になさ*

$$\begin{aligned} n+1 &:= (n_1 - 1, n_2 + 1) & (n_1 > 0 \text{ のとき}) \\ &:= (n_2 + 1, 0) & (n_1 = 0 \text{ のとき}) \end{aligned}$$

$<_o$: 半順序と呼ばれ、次のように定義される。

【0022】 $m_1 \leq n_1, m_2 \leq n_2$ のとき、かつそのときに限って $m \leq_o n$ 。

【0023】 $m <_o n$ は $m \leq_o n$ かつ $m \neq n$ を意味する。

【0024】 $\Sigma_{t^p} := \{m \in \Sigma \mid t \leq_o m, m <_r p\}$
u : u は大きさ q の有限部分2次元配列であり、 Σ_{o^q} から体Kへの写像として定義される。

【0025】F : 体K上の2変数多項式を

【0026】

【外1】

$$f = \sum_{m \in \Sigma} f_m \cdot z^m$$

【0027】とする。

【0028】ただし、 $z^m = x^{m_1} \cdot y^{m_2}$, $\Gamma f = \{m \in \Sigma \mid f_m \neq 0\}$, $s = LP(f) = \max \{m \mid m \in \Gamma f\}$

多項式の組を $F = \{f^{(0)}, \dots, f^{(l-1)}\}$ と表す。

【0029】 $df_n^{(i)} : \Sigma$ 上の点 n から体Kへの写像を u_n とすると、 $LP(f^{(i)}) = s^{(i)}$ なる多項式 $f^{(i)}$ に対し、

【0030】

※

$$\Delta F = \Sigma / \bigcup_{k=1}^l \Sigma s^{(k)}, \quad \Sigma s^{(k)} = \{m \in \Sigma \mid s^{(k)} \leq_p m\}.$$

【0038】 Δ : 上の条件を満たすときの ΔF 。次のようにして求められる。

【0039】 $\Delta = U \Delta q^{-s_o}$, $q, s \in \Sigma_{o^p}$

ただし、 $df_q \neq 0$ かつ $LP(f) = s$ となる $f \in V(u^q)$ が存在するとき

$\Delta^{q-s} = \{m \in \Sigma \mid m \leq_o q - s\}$ 、そうでなければ $\Delta^{q-s} = \emptyset$ 。

【0040】 $h_1 : \text{型} \langle i, j \rangle$ の h_1 を次のように定義する。

【0041】 $h_1 = z^{r-s^{(i)}} \cdot f^{(i)} - (d_i/d_j) \cdot z^{r-n+m-t^{(j)}} \cdot g^{(j)}$

ただし、 $r = (r_1, r_2)$, $r_1 = \max \{s_1^{(i)}, n_1 - s_1^{(j)} + 1\}$, $r_2 = \max \{s_2^{(i)}, n_2 - s_2^{(j)} + 1\}$

*れているので省略する。

【0018】準備1（詳細は文献[8] 参考）：

Σ : あらゆる非負整数 n_1, n_2 の対 $n = (n_1, n_2)$ の集合。

【0019】 n : 点と呼ばれ、 $X-Y$ 平面上の座標 (n_1, n_2) を持つ点と同一視される。

【0020】 $<_r$: 全順序と呼ばれ、 Σ 上の点 n の大小関係を定める。ここでは、点 $n = (n_1, n_2)$ に対し、 $<_r$ に関する次の点を以下のように定める。

【0021】

※【外2】

$$df_n^{(i)} = \sum_{m \in \Sigma} f_m^{(i)} \cdot u^{m+n-s}$$

【0031】とする。

【0032】 $V(u) : p \leq_r q$ に対する u_n の集合を $u^p = \{u_n \mid n \in \Sigma_{o^p}\}$ とするとき、 u^p に対して $df_n^{(i)} = 0$ ($0 \leq n < p$) であれば、 $f[u^p] = 0$ と表す。このとき、 $V(u^p) = \{f \text{ は多項式} \mid f[u^p] = 0\}$ とする。

【0033】定義点：多項式の組Fが与えられた2次元配列を生成することができる極小な多項式系であるとき、 $LP(f^{(i)}) = s^{(i)}$ を定義点と呼ぶ。ただし、極小な多項式系とは次の条件を満たすものである。

【0034】i) $F \subseteq V(u^p)$

ii) $1 \leq i, j \leq l$ かつ $i \neq j$ かつ $LP(f^{(i)}) \geq_p LP(f^{(j)})$ となる i, j は存在しない。

30 【0035】iii) $g \in V(u)$ かつ $LP(g) \in \Delta F$ となる多項式 g は存在しない。

【0036】ただし、

【0037】

【外3】

※

$(j+1) + 1\}$, $t^{(j)} = LP(g^{(j)})$, $f^{(i)} \in V(u^n)$, $g^{(j)} \in V(u^n)$, $d_i = df_n^{(i)}$, $d_j = dg_n^{(j)}$ 。

40 G : Fの補助多項式系と呼ばれ、 $G = \{g^{(0)}, \dots, g^{(l-2)}\}$ なる多項式の組。

【0042】補題1 : ①で定まる型の j は、 $LP(f^{(j)}) = s^{(j)}$ かつ $p - s^{(i)} \leq_p (s_1^{(j)} - 1, s_2^{(j+1)} - 1)$ となる j である。

【0043】補題2 : ②で定まる型の k は、 $LP(f^{(k)}) <_o t$ となる k である。

【0044】阪田アルゴリズム

1) $n = (0, 0)$, $F = \{1\}$, $G = \emptyset$

2) Fの全ての多項式に対して $df_n^{(i)}$ を計算する。

50 【0045】3) もし $df_n^{(i)} \neq 0$ となる $f^{(i)}$ があ

れば、新しい Δ と定義点 t を定める。

4) そのあらゆる定義点 t において以下の手続きを実行する。

【0046】① $t = (s1^{(i)}, s2^{(i)}) \rightarrow$ 補題1の多項式 $h1$ を作る。

【0047】② $t = (n1 - s1^{(i)} + 1, n2 - s2^{(i)} + 1) \rightarrow$ 補題2の型 $\langle k, i \rangle$ の $h1$ を作る。

【0048】③ $t = (n1 - s1^{(i)} + 1, s2^{(j)})$, $1 \leq i \leq l-1 \rightarrow$ 型 $\langle j, i \rangle$ の $h1$ を作る。

【0049】④ $t = (s1^{(i)}, n2 - s2^{(j)} + 1)$, $2 \leq j \leq l \rightarrow$ 型 $\langle i, j-1 \rangle$ の $h1$ を作る。

【0050】⑤ $t = (n1 + 1, s2^{(j)}) \rightarrow h1 = x^{n1-s1^{(j)}+1} \cdot f^{(j)}$

⑥ $t = (s1^{(i)}, n2 + 1) \rightarrow h1 = y^{n2-s2^{(i)}+1} \cdot f^{(i)}$

Fから $df_n^{(i)} \neq 0$ の全ての多項式を除去し、新しく求めた $h1$ を全てFに挿入する。

【0051】5) Δ が変化した場合、新しいFの補助多項式系Gの多項式を旧のGと除去したFの多項式の中から選び出す。

【0052】6) $n = n + 1$; $n = p$ ならば終了; さもないければ2)へ

以下に、阪田アルゴリズムと同じ出力を導出する、本発明による新しいアルゴリズムを示す。

【0053】アルゴリズム1

1) $n = (0, 0)$, $F = \{1\}$, $G = \emptyset$, $A = \{x-1\}$, $B = \{u\}$

2) Bの全ての多項式の最高次係数が0でなければ、新しい Δ と定義点 t を定める。

【0054】3) そのあらゆる定義点 t において以下の手続きを実行する。

【0055】① $t = (s1^{(i)}, s2^{(i)}) \rightarrow$ 補題1の型をもつ多項式 $h2$ を作る。

【0056】② $t = (n1 - s1^{(i)} + 1, n2 - s2^{(i)} + 1) \rightarrow$ 補題2の型 $\langle k, i \rangle$ の $h2$ を作る。

【0057】③ $t = (n1 - s1^{(i)} + 1, s2^{(j)})$, $1 \leq i \leq l-1 \rightarrow$ 型 $\langle j, i \rangle$ の $h2$ を作る。

【0058】④ $t = (s1^{(i)}, n2 - s2^{(j)} + 1)$, $2 \leq j \leq l \rightarrow$ 型 $\langle i, j-1 \rangle$ の $h2$ を作る。

【0059】⑤ $t = (n1 + 1, s2^{(j)}) \rightarrow h2 = x^{-(n1-s1^{(j)}+1)} \cdot b^{(j)}$

⑥ $t = (s1^{(i)}, n2 + 1) \rightarrow h2 = y^{n2-s2^{(i)}+1} \cdot b^{(i)}$

Bから最高次係数が0でない全ての多項式を除去し、新しく求めた $h2$ を全てBに挿入する。

【0060】4) 3)で決定された型の多項式 $h1$ を作る。また、FからBの除去した多項式に対応する多項式を除去し、新しく求めた $h1$ をFに挿入する。

【0061】5) Δ が変化した場合、新しいFの補助多項式系Gの多項式を旧のGと除去したFの多項式の中

ら選び出す。また、新しいBの補助多項式系Aの多項式を旧のAと除去したBの多項式の中から選び出す。

【0062】6) $n = n + 1$; $n = p$ ならば終了; さもないければ2)へ

ただし、

$A, B: A = \{a^{(0)}, \dots, a^{(l-2)}\}$, $B = \{b^{(0)}, \dots, b^{(l-1)}\}$ となる多項式の組。

【0063】 $h2$: 型 $\langle i, j \rangle$ の $h2$ を次のように定義する。

【0064】 $h2 = z^{r-s(i)} \cdot b^{(i)} - (d_i/d_j) \cdot z^{r-n+m-t(j)} \cdot a^{(j)}$

アルゴリズム1と阪田アルゴリズムの大きな違いは、阪田アルゴリズムが $df_n^{(i)}$ を直接計算し、それが全て0であるかどうかを検査するのに対し、アルゴリズム1は $df_n^{(i)}$ を直接計算せず、Bに属す多項式 $b^{(i)}$ の最高次係数が0であるかどうかを検査することであり、そのためにアルゴリズム1は阪田アルゴリズムにない多項式の組A, Bを導入している。

【0065】アルゴリズム1によって得られるFと阪田アルゴリズムによって得られるFが同じ多項式系であることは次のようにして証明できる。

【0066】定理1

式(1)のように定義した u_n ($n \in \Sigma 0p$)を係数とする多項式 u と $LP() = s$ である多項式 f の積多項式を b とするとき、多項式 b の z^{v-n+s} 次の係数 b_{v-n+s} は式(2)のようになる。ただし、 v は任意の整数。

【0067】

$$u = \sum_{n \in \Sigma_0^p} u_n \cdot z^{v-n} \quad (1)$$

$$b_{v-n+s} = \sum_{m \in \Gamma f} f_m \cdot u_{m+n-s} \quad (2)$$

証明: 式(1)から

【0068】

【外4】

$$b = f \cdot u = \sum_m \sum_i f_m \cdot u_i \cdot z^{v-i+m}$$

【0069】である。

【0070】よって、多項式 b の z^{v-n+s} の係数 b_{v-n+s} は $v-i+m = v-n+s$ より $i = m+n-s$ となるので、式(2)のようになる。

【0071】(証明終了) この定理から $f(i) \in F$ に対応する $b(i) = f^{(i)} \cdot u$ の z^{v-n+s} の係数は阪田アルゴリズムの $df_n^{(i)}$ となっていることがわかる。

【0072】また、 $V(u^n)$ である多項式 $f^{(i)}$ を $f_n^{(i)}$ と表すと次の定理が成り立つ。

【0073】定理2: 式(3), (4)によって更新される多項式を $l_q^{(k)}$, $c_q^{(k)}$ とする($q > r, n > r, m$)。また、多項式 $w_q^{(k)}$ を式(5)のように定義すると、 $w_q^{(k)}$ は式(6)によって更新できる。ただし、 r, s, r, t は任意の整数。また、 $l_n^{(i)}$, $l_n^{(j)}$ 及び c_n

(1), $c_m^{(j)}$ の初期値は任意の多項式。

【0074】

$$l_q(k) = z^{rs} \cdot l_n^{(i)} - (d_i / d_j) \cdot z^{rt} \cdot l_m^{(j)} \quad (3)$$

$$c_q(k) = z^{rs} \cdot c_n^{(i)} - (d_i / d_j) \cdot z^{rt} \cdot c_m^{(j)} \quad (4)$$

$$w_q(k) = l_q(k) \cdot e - c_q(k) \cdot d \quad (d, e \text{ は任意の多項式}) \quad (5)$$

$$w_q(k) = z^{rs} \cdot w_n^{(i)} - (d_i / d_j) \cdot z^{rt} \cdot w_m^{(j)} \quad (6)$$

証明：式(6)は任意の q 及び k に対して成り立つので、次が成り立つ。

$$\begin{aligned} \text{【0075】 } w_n^{(i)} &= l_n^{(i)} \cdot e - c_n^{(i)} \cdot d \\ w_m^{(j)} &= l_m^{(j)} \cdot e - c_m^{(j)} \cdot d \end{aligned}$$

従って、次が言える。

$$\begin{aligned} \text{【0076】 } w_q(k) &= l_q(k) \cdot e - c_q(k) \cdot d \\ &= (z^{rs} \cdot l_n^{(i)} - (d_i / d_j) \cdot z^{rt} \cdot l_m^{(j)}) \cdot e \\ &\quad - (z^{rs} \cdot c_n^{(i)} - (d_i / d_j) \cdot z^{rt} \cdot c_m^{(j)}) \cdot d \\ &= z^{rs} \cdot w_n^{(i)} - (d_i / d_j) \cdot z^{rt} \cdot w_m^{(j)} \end{aligned}$$

(証明終了) 阪田アルゴリズムにおいて、 $V(u^q)$ となる $f^{(k)} = f_q^{(k)} \in F$ は $h1$ によって更新され、 $g^{(j)}$ は $V(u^q)$ となる $f^{(j)} = f_m^{(j)}$ であるので、 $h1$ は式(3)の形式で表現できる。

【0077】また、次の定理が成立する。

【0078】定理3：式(5)において、 $l_q(k) = f_q^{(k)}$ 、 $e = u$ 、 $d = z^v \cdot x$ 、及び $c_{-1}^{(j)} = 1$ 、 $c_0^{(i)} = 0$ とすると、 $\deg w_q(k) \leq v$ において $w_q(k) = b_q^{(k)}$ である。このとき、 $b_q^{(k)}$ は $v - q + s^{(k)}$ 次の多項式である。

【0079】証明： $l_q(k) = f_q^{(k)}$ 、 $e = u$ より、式(5)は $w_q(k) = b_q^{(k)} - c_q(k) \cdot d$ となる。また、 $\deg c_{-1}^{(j)} \leq 0$ 、 $\deg c_0^{(i)} \leq 0$ であれば式(4)より $c_q(k)$ は正の次数の多項式であり、 $d = z^v \cdot x$ であるので $\deg c_q(k) \cdot d > v$ である。よって、 $\deg w_q(k) \leq v$ において $w_q(k) = b_q^{(k)}$ である。

【0080】また、定義から $b_q^{(k)} = u \cdot f_q^{(k)}$ の最高次数は $v + s^{(k)}$ であるが、このときの $f_q^{(k)}$ は $f_q^{(k)} \in V(u^q)$ であるので $df_i^{(k)} = 0$ ($i=0, \dots, q-1$)である。定理1から $df_i^{(k)}$ は $b_q^{(k)}$ の $z^{v-i+s^{(k)}}$ の係数であるので、 $b_q^{(k)}$ の $v-i+s^{(k)}$ 次 ($i=0, \dots, q-1$) の係数は0になり、 $b_q^{(k)}$ の最高次数は $v - q + s^{(k)}$ になる。

【0081】(証明終了) 従って、 $b_q^{(k)} = w_q(k)$ は式(6)によって更新できる。即ち、 $b_q^{(k)}$ は $h2$ によって更新できる。この場合、定理1、定理3から d_i 、 d_j は各々多項式 $b_n^{(i)}$ 、 $b_m^{(j)}$ の $z^{v-n+s^{(i)}}$ 、 $z^{v-m+s^{(j)}}$ の係数、即ち、最高次係数である。

【0082】以上から、アルゴリズム1によって得られる F と阪田アルゴリズムによって得られる F が同じ多項式系であることが証明された。阪田アルゴリズムによって得られる多項式系 F は与えられた2次元配列 u を生成する最小多項式系であることが文献[8]によって証明されているので、アルゴリズム1から得られる多項式系 F も与えられた2次元配列 u を生成する最小多項式系であることがいえる。

【0083】よって、このアルゴリズムは例えば図1のような装置によって実現できる。まず、アルゴリズム1の1)で設定されている初期値及びその更新値を記憶するメモリと、そこに格納された B の多項式の最高次係数が0であるかどうかを判断し、0であれば6)を実行し、0でなければ新しい定義点を計算し型を判断する2)を実行する制御回路11と、その型に従って3)、4)に示された $h2$ 、 $h1$ の演算を行う処理回路12によって実現できる。また、制御回路11は定義点が更新されていればメモリ13中の A 、 G の多項式を5)に示されるように新しく選択・更新する。

【0084】ただし、制御回路11と処理回路12は、各種制御手順及び前記アルゴリズムに対応するプログラムをCPUに実行させることによりソフトウェア的に実現することもできるので、分離してある必要はない。また、制御・処理において必要な演算は簡単な整数の乗除算と加減算であるので、特殊な回路・処理は必要としない。また、制御における型の判定及び手順の流れは、予めプログラミングしておいたりROMに焼き付けておいて、必要なときに検索(テーブル・ルックアップ)すれば簡単である。

【0085】また、 $h1$ と $h2$ は同時に実行することができるので処理回路を $h1$ と $h2$ 独立に持つこともできる。また、 $h1$ と $h2$ は同様の処理であるので1つの回路によって実行することとしてもよい。また、効果の所で後述するように本アルゴリズムの特徴は並列処理のしやすさであるので、処理回路は1つである必要はない。以上から、アルゴリズム1を実行する回路は簡単に実現することができることがわかる。

【0086】[実施例2] 次に神谷らに示された Modified decoding algorithm に適したアルゴリズムと等価なアルゴリズムを考える。まず、以下の準備の後に、神谷らによって示されたアルゴリズムを示す。ただし、以下の準備において述べられていない部分は実施例1の場合と同じである。

【0087】準備2 (詳細は文献[12]参照) :

$< r$: ここでは全順序は次のように定める。

$$\text{【0088】 } Q(i) = a \cdot i_1 + b \cdot i_2 \quad (a, b \text{ は互いに素な自然数, } i = (i_1, i_2))$$

$$Q(m) < Q(n) \text{ が満たされるか, } Q(m) = Q$$

(n) かつ $m_2 \leq n_2$ の時に限って、 $m \leq r \cdot n$ とし、この全順序 $< r$ に関して m の次の点を $m+1$ と表す。

$$\text{【0089】 } \Sigma_t^p(n) := \{m \in \Sigma \mid m_2 \leq n, t \leq p, m, m < r \cdot p\}$$

50 u : u は大きさ q の有限部分2次元配列であり、 Σ_0^q

11

(2・(a-1)) から体Kへの写像として定義される。

【0090】F : 体K上の2変数多項式を

【0091】

【外5】

$$f = \sum_{m \in \Gamma f} f_m \cdot z^m$$

【0092】とする。

【0093】ただし、 $z^a = x^{a_1} \cdot y^{a_2}$, $\Gamma f = \{m \in \Sigma(a-1) \mid f_m \neq 0\}$, $s = LP(f) = \max \{m \mid m \in \Gamma f\}$

多項式の組を $F = \{f^{(0)}, \dots, f^{(a-1)}\}$ と表す。

【0094】 $V(u) : Q(p) < q$ に対する u_n の集合を $u^p = \{u_n \mid n \in \Sigma_0^p(2 \cdot (a-1))\}$ とするとき、 u^p に対して $df_n^{(1)} = 0$ ($n \in \Sigma_0^p(a-1+s_2)$) であれば、 $f[u^p] = 0$ と表す。このとき、 $V(u^p) = \{f \text{ は多項式} \mid f[u^p] = 0\}$ とする。

【0095】 Δ : Δ は次のようにして求められる。

【0096】 $\Delta = \bigcup \Delta^{q-s}$

$q, s \in \Sigma_0^p(2 \cdot (a-1))$

ただし、 $df_a \neq 0$ かつ $LP(f) = s$ となる $f \in V(u^q)$ が存在するとき $\Delta^{q-s} = \{m \in \Sigma(a-1) \mid m_1 \leq q_1 - s_1, m_2 = q_2 - s_2\}$ 、そうでなければ $\Delta^{q-s} = \emptyset$ 。

【0097】 h_3 : 型 $\langle i, j \rangle$ の h_3 を次のように定義する。

【0098】 $h_3 = x^{r_1-s_1(i)} \cdot f^{(i)} - (d_i/d_j) \cdot x^{r_1-n_1+m_1-t_1(j)} \cdot g^{(j)}$

$G : F$ の補助多項式系と呼ばれ、 $G = \{g^{(0)}, \dots, g^{(a-1)}\}$ なる多項式の組。

神谷アルゴリズム

1) $n = (0, 0)$, $F = \{1, y, y^2, \dots, y^{a-1}\}$, $G = \emptyset$

2) F の全ての多項式に対して $df_n^{(1)}$ を計算する。

【0099】3) もし $df_n^{(1)} \neq 0$ となる $f^{(i)}$ があれば、新しい Δ と定義点 t を定める。

4) そのあらゆる定義点 t において以下の手続きを実行する。

【0100】 $t = (s_1^{(1)}, s_2^{(1)}) \rightarrow$ 型 $\langle i, n_2 - i \rangle$ の h_3 を作る。

【0101】 $t = (n_1 - s_1^{(1)} + 1, n_2 - s_2^{(1)}) \rightarrow$ 型 $\langle n_2 - i, i \rangle$ の h_3 を作る。

【0102】 $t = (n_1 + 1, n_2 - s_2^{(1)}) \rightarrow h_3 = x^{n_1-s_1(n_2-1)+1} \cdot f^{(n_2-1)}$

F から $df_n^{(1)} \neq 0$ の全ての多項式を除去し、新しく求めた h_3 を全て F に挿入する。

【0103】5) Δ が変化した場合、新しい F の補助多項式系 G の多項式を旧の G と除去した F の多項式の中から選び出す。

【0104】6) $n = n + 1$; $n = p$ ならば終了; さもなければ 2) \rightarrow

12

神谷アルゴリズムと等価なアルゴリズムを以下に示す。

【0105】アルゴリズム 2

1) $n = (0, 0)$, $F = \{1, y, \dots, y^{a-1}\}$, $G = \emptyset$, $A = \{x-1\}$, $B = \{u\}$

2) B の全ての多項式の最高次係数が 0 でなければ、新しい Δ と定義点 t を定める。

【0106】3) そのあらゆる定義点 t において以下の手続きを実行する。

【0107】① $t = (s_1^{(1)}, s_2^{(1)}) \rightarrow$ 型 $\langle i, n_2 - i \rangle$ の h_4 を作る。

【0108】② $t = (n_1 - s_1^{(1)} + 1, n_2 - s_2^{(1)}) \rightarrow$ 型 $\langle n_2 - i, i \rangle$ の h_4 を作る。

【0109】③ $t = (n_1 + 1, n_2 - s_2^{(1)}) \rightarrow h_4 = x^{s_1(n_2-1)-n_1-1} \cdot f^{(n_2-1)}$ B から最高次係数が 0 でない全ての多項式を除去し、新しく求めた h_4 を全て B に挿入する。

【0110】4) 3) で決定された型の多項式 h_3 を作る。また、 F から B の除去した多項式に対応する多項式を除去し、新しく求めた h_3 を F に挿入する。

【0111】5) Δ が変化した場合、新しい F の補助多項式系 G の多項式を旧の G と除去した F の多項式の中から選び出す。また、新しい B の補助多項式系 A の多項式を旧の A と除去した B の多項式の中から選び出す。

【0112】6) $n = n + 1$; $n = p$ ならば終了; さもなければ 2) \rightarrow

ただし、

h_4 : 型 $\langle i, j \rangle$ の h_4 を次のように定義する。

【0113】 $h_4 = x^{r_1-s_1(i)} \cdot b^{(i)} - (d_i/d_j) \cdot x^{r_1-n_1+m_1-t_1(j)} \cdot a^{(j)}$

30 アルゴリズム 2 と神谷アルゴリズムの違いも、アルゴリズム 1 と阪田アルゴリズムの違いと同様に $df_n^{(1)}$ を直接計算するか、 B に属す多項式 $b^{(i)}$ の最高次係数とするかである。従って、アルゴリズム 1 と阪田アルゴリズムの間で成り立った関係が、アルゴリズム 2 と神谷アルゴリズムの間で成立すればアルゴリズム 2 によって得られる F と神谷アルゴリズムによって得られる F が同じ多項式系であることが証明できる。よって、次のようになる。

【0114】定理 1 の $f^{(i)} \in F$ を神谷アルゴリズムの $f^{(i)}$ とすれば、それに対応する多項式 $b^{(i)} = f^{(i)} \cdot u$ の z^{v-n+s} の係数は神谷アルゴリズムの $df_n^{(1)}$ となっていることがいえる。よって、定理 1 はこの実施例においても成立する。

【0115】定理 2 は一般的に成立する。

【0116】定理 3 の $f_a^{(k)}$ は神谷アルゴリズムの $f^{(i)}$ であるので、定理 3 も同様に成立する。

【0117】従って、阪田アルゴリズムの場合と同様に $w_a^{(k)} = b_a^{(k)}$ であることがいえるので、 $b_a^{(k)}$ は h_4 によって更新できる。この場合も、定理 1 から d_i , d_j は各々多項式 $b_n^{(i)}$, $b_m^{(k)}$ の最高次数の係数であ

る。

【0118】以上から、アルゴリズム2によって得られるFと神谷アルゴリズムによって得られるFが同じ多項式系であることが証明された。

【0119】よって、アルゴリズム2も図1のような装置によって実現できる。ただし、制御及び処理はアルゴリズム1と異なる。(処理はh3, h4の演算を行い、制御は型の判定が簡単化されている。また、メモリ部もA, B, F, Gの初期値も1)に示された多項式が設定される。)また、アルゴリズム2も並列処理に適しているので、処理回路は1つである必要はない。以上から、アルゴリズム2を実行する回路も簡単に実現することができる。

【0120】【実施例3】多次元配列を生成する記憶素子数最小の多次元シフトレジスタを合成する多項式系Fを導出するアルゴリズムもまた、阪田によって示されている[13]。このアルゴリズムは多次元BM法と呼ばれており、ここでは、多次元BM法に対応する新しいアルゴリズムを考える。

【0121】詳細なアルゴリズムは煩雑であるので、BM法と呼ばれるアルゴリズムの特徴を図で示す。1次元のBM法を含め、その拡張である阪田によって提案されたアルゴリズム(多次元BM法)は図2のような構成を持っている。それに対応するアルゴリズムは図3のようになる。図2と図3の違いも前述の2つの実施例と同様に、 $df_n^{(1)}$ を直接計算するか、Bに属す多項式 $b^{(1)}$ の最高次係数とするかである。従って、定理1~3が多次元配列においても成立すれば、図2の多次元BM法によって得られるFと図3のアルゴリズムによって得られるFが同じ多項式系であることが証明できる。

【0122】前述の実施例において点は $n = (n_1, n_2)$ となる2次元空間上で定義され、配列及び多項式の各係数はその点における写像である。これをN次元空間に拡張した場合、点は $n = (n_1, n_2, \dots, n_N)$ と定義され、配列及び多項式もその点における写像になり、その関係は次元を問わず同じである。従って、N次元空間においても定理1~3は同様に成り立つことは明らかである。

【0123】従って、図3のアルゴリズムから生成されるFは図2のアルゴリズムから生成されるFと同じであることがいえる。よって、図3のアルゴリズムも図1のような装置によって簡単に実現できる。

【0124】ただし、図3のアルゴリズムは1次元の場合*

$$b_{n+1}^{(k)} = z^{rs} \cdot b_n^{(1)} - (d_i / d_j) \cdot z^{rt} \cdot a_n^{(j)} \quad (7)$$

$$b_{n+1}^{(k)} = d_j \cdot z^{rs} \cdot b_n^{(1)} - d_i \cdot z^{rt} \cdot a_n^{(j)} \quad (8)$$

以上からわかるように、Eu法とBM法の違いを次のように表せる。

【0132】BM法： $df_n^{(1)}$ を直接計算して、その値を用いて $f_{n+1}^{(k)}$ を求める。

【0133】Eu法： $df_n^{(1)}$ を直接計算せず、多項式

*合、前述のRS符号やBCH符号の復号において用いられるEu法と呼ばれるアルゴリズムになる。また、効果の所で述べるように図3のアルゴリズムは1次元Eu法の特徴を失うことなく、多次元へとEu法を拡張している。従って、アルゴリズム1, 2を含む図3の構成を持つアルゴリズムを多次元BM法に対応して多次元Eu法と呼ぶことにする。

【0125】【その他の実施例】前述したように1次元のBM法や1次元のEu法はRS符号やBCH符号においてよく用いられているので、種々の変形アルゴリズムが提案されている。よく知られた1次元Eu法の変形アルゴリズムとして連分数法(L. R. Welch and R. A. Scholtz: "Continued fractions and Berlekamp's algorithm," IEEE Trans. Inf. Theory, IT-25, pp. 19-27, Jan. 1979)がある。これは、式(6)の演算を最後まで行わず、最終結果に関係のない演算を省いた手法である。しかし、式(6)の形式の演算によってBを更新し $df_n^{(1)}$ を直接計算しないという点では同じであるので、本方式は連分数法を含む種々のアルゴリズムの変形に対しても有効であることは明かである。

【0126】また、本方式はアルゴリズム1, 2の例からもわかるように型の分類の方法に依存しない。従って、本方式は多次元配列に対してBM法のように直接 $df_n^{(1)}$ を計算せず、式(6)の形式の演算によってBを更新し、 $df_n^{(1)}$ をBの多項式の最高次係数として演算する手法に対して全て有効である。

【0127】また、点や各パラメータ(r_s, r_t 等)は整数とし、 $f_n^{(1)}$ 及び $b_n^{(1)}$ は多項式としているが、点や各パラメータを任意の複素数、多項式を有理関数等に拡張しても本方式が有効であることは明かである。

【0128】また、多項式uやfの次数を逆(双対多項式)にすれば、多項式 $b = f \cdot u$ も次数が逆になり d_i を多項式bの最低次係数とすることもできる。

【0129】また、定理3において $d = z^v \cdot x$ としているが、 $d = z^{v+1}$ または $d = 0$ としても $w_q^{(k)} = b_q^{(k)}$ となり図3の構造を持つアルゴリズムの結果は変わらないことは明かである。

【0130】また、以上の実施例ではBの更新のために次の式(7)のような演算を用いているが、式(8)のような演算によっても定数項の違いを除いて同じ結果が得られる。ただし、 $a_n^{(j)} = b_m^{(j)}$ 。

【0131】

$b_n^{(1)}$ と $b_n^{(j)}$ の最高次の係数 d_i, d_j を用いて式(7)のように $b_{n+1}^{(k)}$ を求める。また、同じ d_i, d_j を用いて $f_{n+1}^{(k)}$ を計算する。ただし、 $a_n^{(j)} = b_m^{(j)}$ 。

【0134】

15

$$b_{n+1}(k) = z^{ra} \cdot b_n(i) - (d_i / d_j) \cdot z^{rt} \cdot a_n(j)$$

この違いから次のような相違点がBM法とEu法の装置化及び高速化に対して生じる。

【0135】BM法： BM法は $f_n(i)$ と $f_{n+1}(k)$ を並列に計算することは困難である。なぜならば、 $f_{n+1}(k)$ を計算するためには $df_n(i) = d_i$ が必要であり、 $df_n(i)$ は $f_n(i)$ の全ての係数を用いて計算される。従って、 $df_n(i)$ が得られるのは $f_n(i)$ の計算が終わった後であるので、 $f_{n+1}(k)$ の計算は $f_n(i)$ の計算が終わった後でしか始まらない。従って、図4(a)に示すようにBM法では点nとn+1における多項式 $f_n(i)$ と $f_{n+1}(k)$ は逐次的に計算される。さらに、BM法は $f_n(i)$ と $f_{n+1}(k)$ 及び $df_n(i)$ の演算を1クロックで計算したとしても、それらが並列に実行されないため図4(b)に示すように2pクロック程度の計算時間が必要である。これは、1次元BM法から図2に示される多次元BM法に対して共通する特徴である。

【0136】Eu法： Eu法は $f_n(i)$ と $f_{n+1}(k)$ を並列に計算することができる。なぜならば、Eu法において d_i は $b_n(i) \in B$ の最高次係数となっており、Bに属す多項式は式(7)によってBに属す多項式のみによって計算される。 $b_n(i)$ と $a_n(j)$ を用いて式(7)が最高次数から下位次数の方へ順次計算されているとすると、出力 $b_{n+1}(k)$ は最高次係数から下位次数の方へ順次得られる。よって、点n+2における多項式 $b_{n+2}(k)$ は $b_{n+1}(i)$ の最高次係数が得られた時点から計算を始めることができる($a_{n+1}(j)$ は既に得られている多項式である。)。これは点n+1とn+2における多項式が並列に計算できることを示しているが、点nとn+1においても同様であることは明かである。従って、Bの更新のための演算は各点において並列に実行することができる。また、 $f_{n+2}(k)$ と $f_{n+1}(k)$ の計算も $b_{n+1}(i)$ と $b_n(i)$ の最高次係数がわかった時点で始めることができるので、図5(a)に示すように $f_{n+1}(k)$ の演算も各点において並列に実行できる。よって、Eu法は式(7)の演算を行う処理回路を複数用意すれば、その数に比例した高速化が容易に行える。これは、よく知られた1次元Eu法から図3の多次元Eu法に対して共通の特徴であ

16

(7)

り、上記のBM法にはない特徴である。また、 $b_n(i)$ 、 $b_{n+1}(k)$ 及び $f_n(i)$ 、 $f_{n+1}(k)$ の演算を1クロックで計算した場合、図5(b)に示すようにpクロック程度の計算時間で良く、BM法より高速な演算が行える(ただし、図5は点nにおける d_i を d_n と表す。)

【0137】一般的に、高速なアルゴリズムとは計算量の少ないアルゴリズムを指す場合が多い。しかし、高速化を実現する場合、アルゴリズムの計算量を減少させるアプローチとアルゴリズムの並列度を高めるアプローチが考えられる。それは並列処理によって複数の処理を同時に実行すれば処理時間が短縮できるためである。最近ではVLSI技術の進歩により大規模な並列処理チップを容易に実現することができる。よって、並列度の高いアルゴリズムの方が高速処理向きである場合も多い。

【0138】

【発明の効果】以上説明したように、本発明によるEu法は、BM法よりも並列度が高く、高速処理に向いていると言える。また、BM法が多項式 $f_n(i)$ とスカラー量 $df_n(i)$ という2種類の演算を必要とするのに対して、Eu法は $f_n(i)$ 、 $b_n(i)$ に対して式(7)に示される同一形式の演算のみでよいので、処理回路をユニット化しやすくより並列処理に向いていると言える。

【0139】従って、並列処理を用いて装置化する場合、本発明によれば、簡単に高速な処理装置を構成できるという効果がある。

【図面の簡単な説明】

【図1】本発明による多次元配列生成回路である。

【図2】従来のBM法による多次元配列生成アルゴリズムである。

【図3】本発明によるEu法による多次元配列生成アルゴリズムである。

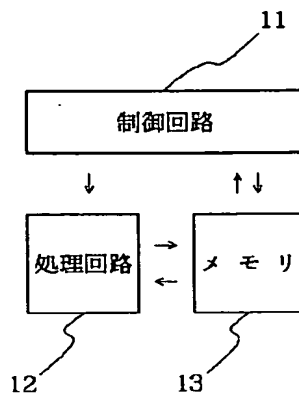
【図4】従来のBM法の動作説明図である。

【図5】本発明によるEu法の動作説明図である。

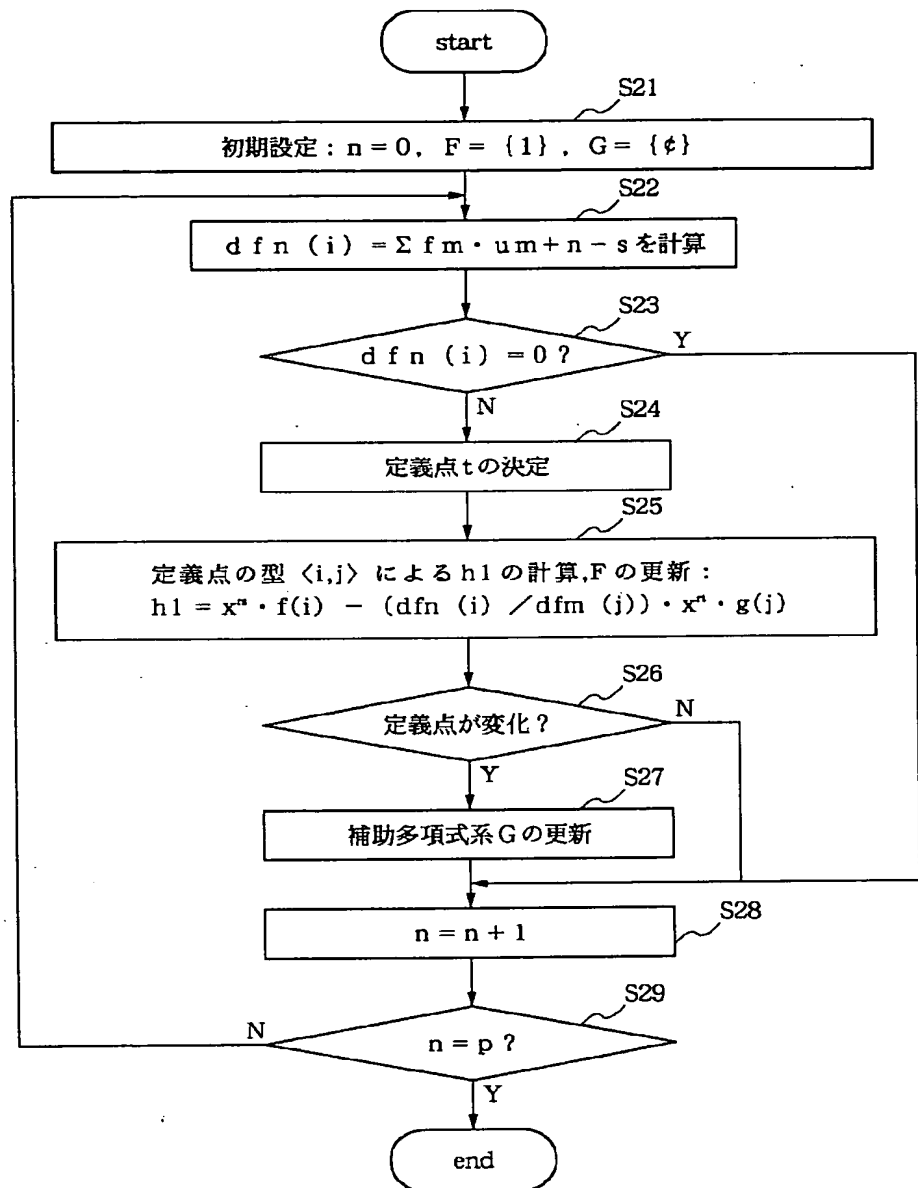
【符号の説明】

- 11 制御回路
- 12 処理回路
- 13 メモリ

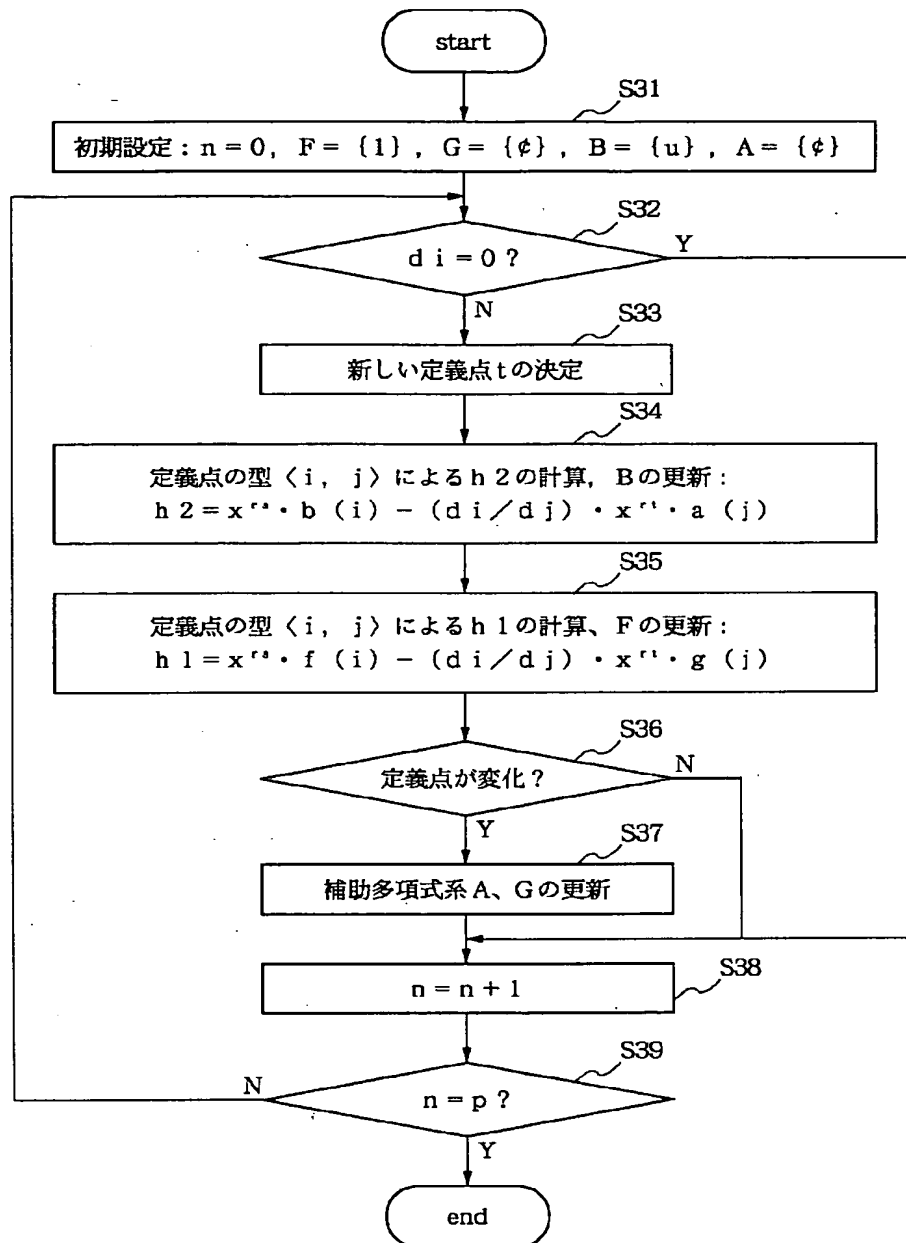
【図1】



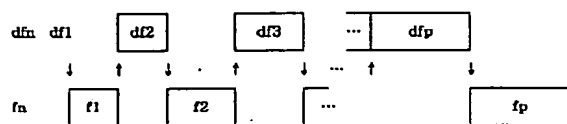
【図2】



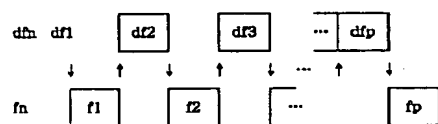
【図3】



【図4】

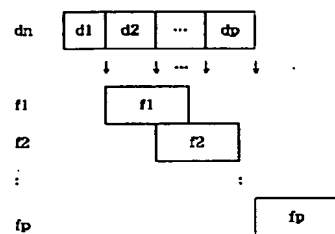


(a)

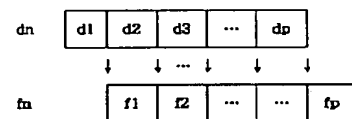


(b)

【図5】



(a)



(b)